

REMARKS

Reconsideration of the application in view of the above amendments and the following remarks is respectfully requested. Claims 1 and 13 have been amended. No claims have been canceled or added. Claims 1-24 are currently pending in the application.

Claim Rejection - 35 U.S.C. §112

In the Office Action, the Examiner rejected claims 1-24 under 35 U.S.C. §112, second paragraph, as being indefinite. Specifically, the Examiner alleged that the term "approximates" recited in claims 1 and 13 is indefinite (not clear what the metes and bounds are). While Applicant disagrees with this allegation, Applicant has nonetheless, in the interest of furthering prosecution, amended claims 1 and 13 to remove this term. Hence, withdrawal of this rejection is respectfully requested.

Claim Rejections – 35 U.S.C. § 102(e)

In the Office Action, the Examiner rejected Claims 1, 2, 4-14, and 16-24 under 35 U.S.C. §102(e) as being anticipated by Sundaresan (U.S. Patent No. 6,487,566). This rejection is respectfully traversed.

Independent claim 1

With regard to independent claim 1, there is recited:

A computer-implemented method for generating a transformation document, comprising:
analyzing a target document, said target document comprising one or more patterns; and
automatically generating, based at least upon said target document, a transformation document which, when processed in conjunction with a source document, causes said source document to be transformed into a result document that comprises said one or more patterns (emphasis added).

Claim 1 provides an advantageous method for automatically generating a transformation document based upon a target document. Typically, transformation documents are produced manually; however, with the method of claim 1, a transformation document may now be generated automatically based upon a target document.

Such a method is neither disclosed nor suggested by Sundaresan. Instead, Sundaresan discloses a method for transforming a source XML document into a result XML document. More specifically, in Sundaresan, a set of rule specifications 114 (Fig. 1) are provided. Based on the rule specifications, Java class specifications 116 are created by a pre-processing component 112. Col. 5, lines 42-45. When instantiated, the Java class specifications 116 give rise to a set of transformation components 118, 120, 122 that make up a web daemon 108. Col. 5, lines 45-50. The set of transformation components then operate on an input XML document 124 to derive an output XML document 126. Col. 5, lines 41-50. In this manner, the source XML document 124 is transformed by the transformation components 118, 120, 122 into a result XML document 126.

In rejecting claim 1, the Examiner stated:

Regarding claims 1 and 13, Sundaresan discloses a computer system and method for generating a transformation document comprising: (fig. 1, 106)

analyzing a target document, said target document comprising one or more patterns; and (fig. 1, item 124, col. 4, lines 35-40 and col. 6, lines 33-37)

automatically generating, based at least upon said target document, a transformation document (fig. 1, item 108/118), which, when processed in conjunction with a source document (fig. 1, item 112 and col. 5, lines 41-50) causes said source document to be transformed into a result document that...comprises said at least one or more patterns (fig. 1, item 126 and col. 6, lines 29-63).

From this excerpt, it is clear that the Examiner is contending the following: (1)

the XML input 124 of Sundaresan is the target document of claim 1; (2) the web daemon 108 and the pattern matching component 118 are the transformation document of claim 1; (3) the web daemon 108 and the pattern matching component 118 are automatically generated based upon the XML input 124; (4) the pre-processing component 112 is the source document of claim 1; (5) the XML output 126 is the result document of claim 1; and (6) the web daemon 108 and pattern matching component 118, when processed in conjunction with the pre-processing component 112, cause the pre-processing component 112 to be transformed into the XML output 126. Applicant respectfully disagrees with these contentions.

First of all, it should be noted that the web daemon 108 and pattern matching component 118 (referred to collectively herein as the web daemon 108) can in no way be interpreted as a "document". Rather, it is a collection of objects that is instantiated at run time to perform a transformation function. Given that the web daemon 108 is a collection of functional components, Applicant cannot see how it can be interpreted as a "transformation document" as that term is used in claim 1.

Even if the web daemon 108 could be interpreted as a transformation document, it is in no way automatically generated based upon the XML input 124, as contended by the Examiner. In Sundaresan, the web daemon 108 (made up of components 118, 120, 122) is used to transform the XML input 124 into the XML output 126. Col. 5, lines 45-50. Thus, rather than being generated based upon the XML input 124, the web daemon 108 operates on the XML input 124 to transform it into the XML output 126. In fact, by the time the XML input 124 is provided to the web daemon 108 and more specifically the pattern matching component 118, the web daemon 108 is already up and running (i.e. existing). Col. 5, lines 45-50. Thus, the web daemon 108 could not possibly be generated based upon the XML input 124. Overall, there is no teaching or suggestion whatsoever in Sundaresan that the web daemon 108 is generated based upon the XML input 124.

In addressing this argument made by Applicant in a previous response, the Examiner stated on page 8 of the Office Action:

Applicants argue on page 11 of the 2/7/05 response that Sundaresan does not disclose or suggest that a transformation document be automatically generated based upon a target or result document. Examiner disagrees. Sundaresan clearly discloses and teaches automatic document transformation, see col. 4, lines 22-24. Sundaresan automatically transforms documents based on the patterns of source and target documents, see fig. 1 and col. 6, lines 29-35. (Emphasis added)

From this excerpt, it appears that the Examiner is misunderstanding the crux of Applicant's argument. Applicant is not contending that Sundaresan does not teach automatic document transformation. To the contrary, Applicant agrees with the Examiner that Sundaresan does teach automatic transformation of a source XML document 124 into a result XML document 126 (see the summary of Sundaresan's operation provided above). However, what Applicant is arguing is that, unlike claim 1,

Sundaresan does not disclose or suggest the automatic generation of a transformation document based upon a target document. Automatic document transformation and automatic generation of a transformation document are two very different things.

Automatic document transformation involves the use of a transformation mechanism (such as a transformation document) to transform a source document into a result document. Automatic generation of a transformation document, on the other hand, involves the automatic generation (based upon a target document) of the transformation document that will be used in the transformation process. The former is taught by Sundaresan. The latter is not. As noted on page 3, lines 4-15, of Applicant's disclosure, the use of a transformation document to automatically transform a source document into a target document is known. What is not known, and what is claimed in claim 1, is the automatic generation of the transformation document (based upon a target document) that will be used in the transformation process. There is nothing in Sundaresan that discloses or suggests that the web daemon 108 (which the Examiner has interpreted to be the transformation document) is automatically generated based upon the XML input 124 (which the Examiner has interpreted to be the target document).

In further addressing this argument made by Applicant in the previous response, the Examiner stated on page 9 of the Office Action:

Last, in Sundaresan system, the target document 124 is clearly input to the transformation document 108/118 as can be seen in the first figure. Applicant's cannot logically argue that the target document 124 is a source document therefore it is different without substantial support and evidence, and even if the target document 124 could be interpreted as a source document Examiner contends that a desired source document (i.e., inputted document 124) is equivalent to a target document because that is the sought, desired, result document.

Again, it appears that the Examiner is misunderstanding Applicant's argument. Applicant is not contending that the XML input 124 is not provided as input to the web daemon 108. It is. However, what Applicant is arguing is that the XML input 124 (the alleged target document) is not provided for the purpose of automatically generating the web daemon 108 (the alleged transformation document). Put another way, the web daemon 108 is not automatically generated based upon the XML input 124. As noted previously, by the time the XML input 124 is provided to the web daemon 108 and more specifically to the pattern matching component 118, the web daemon 108 is already instantiated. Hence, the web daemon 108 could not possibly be generated based upon the XML input 124. Overall, Sundaresan does not disclose or suggest in any way the automatic generation of a transformation document based upon a target document.

Another point to note regarding the Examiner's contentions is that the pre-processing component 112 of Sundaresan (which the Examiner has interpreted to be the source document) cannot be processed in conjunction with the web daemon 108 (the alleged transformation document) to produce the XML output 126 (the alleged result document). In Sundaresan, the pre-processing component 112 is used to derive the Java specifications 116, which in turn, are used to give rise to the web daemon 108. Col. 5, lines 45-50. Thus, rather than being processed in conjunction with the web daemon 108, the pre-processing component 112 is used to give rise to the web daemon 108. Nowhere in Sundaresan is it disclosed or suggested that the pre-processing component 112 be processed in conjunction with the web daemon 108. In fact, if this were done, the result would not in any way resemble the XML output 126 (the alleged result document).

In response to this argument made by Applicant in a previous response, the Examiner stated on page 9 of the Office Action:

Applicants argue on page 13 of the 2/7/05 response that no where in "Sundaresan" is it disclosed or suggested that the pre-processing component 112 be processed in conjunction with the web daemon 108. Examiner disagrees. The pattern matching component (transformation document) 108/118 is derived by the pre-processing component 112 and target input 124 to generate output 126, please refer to fig. 1 and col. 5, lines 41-50.

In the section of Sundaresan (col. 5, lines 41-50) cited by the Examiner, it is stated:

A pre-processing component 112 uses XML-based rule specifications 114 to create pattern match pre-process information 116 and Java class specifications 116. The Java class specifications 116 are then instantiated as one or more pattern matching components 118 that accept and process XML input 124....

This excerpt makes it clear that, in contrast to the Examiner's contentions, the pre-processing component 112 is not processed in conjunction with the pattern matching components 118 (which is a part of the web daemon 108). Rather, the pre-processing component 112 creates the Java class specifications 116, which are then instantiated as the pattern matching components 118. Thus, rather than being processed in conjunction with the pattern matching components 118, the pre-processing component 112 gives rise to the pattern matching components 118. Also, this excerpt makes it clear that it is the XML input 124, not the pre-processing component 112, that is accepted and processed by the pattern matching components 118. Thus, rather than supporting the Examiner's contention, this excerpt refutes it. Overall, there is nothing in Sundaresan that discloses or suggests that the pre-processing component 112 (the alleged source document) may be processed in conjunction with the pattern matching components 118 (the alleged

transformation document) to give rise to the XML output 126 (the alleged result document).

For at least the reasons given above, Applicant respectfully submits that Sundaresan, as interpreted by the Examiner, fails to disclose or suggest the method of claim 1. Consequently, Applicant submits that claim 1 is patentable over Sundaresan.

Applicant further submits that dependent claims 2, and 4-12, which depend from claim 1 and which recite further advantageous aspects of the invention, are likewise patentable over Sundaresan for at least the reasons given above in connection with claim 1.

Claims 13, 14, and 16-24 include limitations similar to claims 1, 2, and 4-12, except in the context of computer-readable media. It is therefore respectfully submitted that claims 13, 14, and 16-24 are patentable over Sundaresan for at least the reasons given above with respect to claims 1, 2, and 4-12.

Claim Rejections – 35 U.S.C. §103(a)

In the Office Action, the Examiner rejected claims 3 and 15 under 35 U.S.C. §103(a) as being unpatentable over Sundaresan in view of Nasr (U.S. Patent No. 6,263,332). This rejection is respectfully traversed.

Like Sundaresan, Nasr discloses a method for applying queries and transformation rules to source documents to derive result documents. More specifically, Nasr teaches “retrieving information in a first markup language through a query engine and presenting the information in any required markup language. A user inputs a query

and may invoke a number of transformation sequences,” which “contain a markup language pattern and an action, which may include transforming the tags” (Abstract).

The Examiner relies on Nasr to show that the transformation document of claim 1 is an XSLT document as recited in claim 3. For the sake of argument, it will be assumed that Nasr discloses the subject matter as contended by the Examiner. Even if this were true, however, the combination of Sundaresan and Nasr still would not produce the method of claim 3. As argued above in connection with claim 1 (from which claim 3 depends), Sundaresan fails to disclose or suggest a number of aspects of claim 1. Nasr also fails to disclose or suggest these aspects. Thus, even if the references were combined (assuming for the sake of argument that it would have been obvious to combine the references), they still would not disclose every element of claim 3. Thus, Applicant submits that claim 3 is patentable over Sundaresan and Nasr.

Claim 15 includes limitations similar to claim 3, except in the context of computer-readable media. It is therefore respectfully submitted that claim 15 is patentable over Sundaresan and Nasr for at least the reasons given above with respect to claim 3.

Conclusion

For the reasons given above, Applicant submits that the pending claims are patentable over the art of record, including the art cited but not applied. Accordingly, allowance of all pending claims is respectfully solicited.

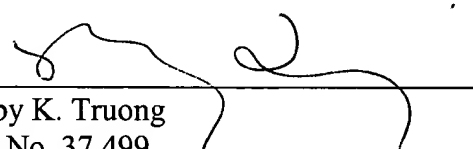
The Examiner is requested to telephone the undersigned attorney at (408) 414-1234 prior to issuance of the next action to discuss any outstanding issues that may advance prosecution.

No fee is believed to be due specifically in connection with this Reply. The Commissioner is authorized to charge any fee that may be due in connection with this Reply to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: July 5, 2005


Bobby K. Truong
Reg. No. 37,499

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 414-1080 ext. 234
Facsimile No.: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

on July 5, 2005
(Date)

by Dani Faisanotes
(Signature)